A 420 GOPS/W CGRA with a Configurable MAC and Dynamic Truncation

*Yi Sheng Chong†, *Rakshith Harish‡, Rajesh Chandrasekhara Panicker‡, Vishnu P. Nambiar†, Anh Tuan Do†
†Institute of Microelectronics, Agency for Science, Technology and Research (A*STAR), Singapore
‡Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore
*equally contributed

***Abstract*—Edge devices demand for highly efficient yet flexible processing capability to handle dynamic real-time workloads. Coarse grain reconfigurable architecture (CGRA) emerges as a suitable accelerator candidate in edge devices, because they are as flexible as general purpose processors and offer high efficiency close to that of domain specific accelerators. However, a typical CGRA requires two cycles for a multiply-and-accumulate (MAC) operation, and workloads such as neural network inference and signal processing involve many MAC operations, resulting in long CGRA processing time. This work proposes a CGRA that has configurable MAC units in the processing elements (PEs) that can perform an addition (ADD) or multiplication (MUL) or a MAC by using the same multiplier and adder, in a single cycle. The readout precision of MAC result can be adjusted by a truncation block. The proposed CGRA is implemented with 40nm CMOS technology. It attains an energy efficiency of 420.6GOPS/W operating at supply of 0.6V and frequency of 21MHz, which is 1.4 times higher than the state-of-the-art.**

I. Introduction

Edge computing plays an important role in transforming the way the data generated on the edge devices is stored, processed and transported [1]. Edge devices such as mobile phones and smartwatches are equipped with sensors to deliver user experience. For example, a smartphone runs image recognition to detect objects in a photo captured using a camera, at the same time detects user voice commands recorded using a microphone. To handle various workloads in real time, an edge device requires flexible yet powerful processing capability. This is achieved by the general purpose processors, which attain high performance when scaled according to Moore's law [2], [3]. However, this has led to the 'power wall' phenomenon [4], compelling chip designers to prioritize energy efficiency over sheer performance to conserve battery power.

As shown in Fig. 1, CGRA offers higher efficiency and flexibility than a general purpose processor and a domain specific accelerator respectively. A typical CGRA consists of a 2D array of processing elements (PEs) that can perform arithmetic, logical and memory operations [5]. The PEs are connected to their neighbours through reconfigurable switches to send or receive data [6]. To run processing on a CGRA, a compiler is employed to generate machine code that includes PE instructions and computation data, to inform the CGRA on the sequence of operations [7]. The compiler maps any given workload to the CGRA, thus the CGRA is said to be flexible. The CGRA can offer high performance since the compiler can exploit parallelism when mapping a workload, where multiple operations can be executed in parallel in multiple PEs [5]. Thus, CGRA is a good accelerator candidate in edge devices as it is programmable to process different workloads with high energy efficiency [8], [9].

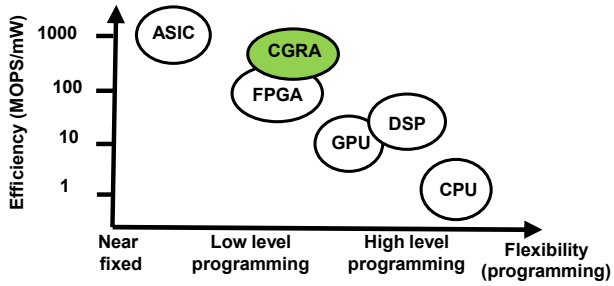

Fig. 1. Spectrum of computing architectures where CGRA is relatively flexible and highly programmable [13]

A drawback of current CGRA architectures is that it typically needs two cycles to perform a multiply-and-accumulate (MAC) operation. A MAC operation is typically mapped to two instructions, i.e., one multiplication (MUL) and one addition (ADD) instructions, such that the MUL and ADD are handled by the multiplier and adder in the PE respectively [10]. The MAC operations are frequently used in many computations, such as neural network for image recognition [11] and fast Fourier transform (FFT) for digital signal processing [12]. A CGRA takes long time to process workloads with high number of MAC operations.

This paper proposes a CGRA PE having a configurable MAC block and a configurable truncation block. The MAC block is configurable to perform either MUL, ADD or MAC operation. The configurable truncation block detects overflow and truncates the accumulation read out to the precision specified by the user. As a result, the CGRA PE reduces the number of cycles when processing workloads that have MAC operations. Our results show improvement in overall CGRA energy efficiency of up to 37% when running the general matrix multiplication (GeMM) workload.

The rest of this paper is organized as follows. Section II provides a brief discussion of the proposed CGRA and an in-depth description of the PE, configurable MAC and truncation block in the PE. Section III presents the performance evaluation methodology and implementation results, followed by the conclusions in Section IV.

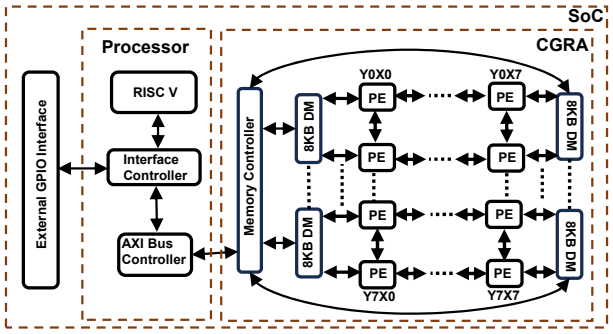

Fig. 2. Proposed CGRA that is integrated with a RISC-V based CPU in a system-on-chip

II. Proposed CGRA

A. Structure of the CGRA

Fig. 2 presents the proposed CGRA that is based on [14]. The CGRA is integrated in a system-on-chip where a RISC-V based CPU is used to coordinate the control and communication needs. After a CGRA workload is compiled, the CGRA is updated with a new set of PE instructions and computation data via the the CPU. The proposed CGRA has an 8x8 PE array and data memory (DM) of 64KB. The DM stores the computation data. The PE are meshed connected, where the PE can communicate up to 4 neighbouring PEs. The edge PEs are connected to the DM to receive and send computation data. The PE design is elaborated in Section II-B.

B. Structure of the PE

Fig. 3(a) shows the PE architecture consisting of a configuration memory (CM), arithmetic logic unit (ALU), a control unit and a crossbar router. The control unit fetches and decodes the 64-bit instruction from CM. The instruction configures the ALU operation, and switches in the router. The ALU is capable of performing 18 arithmetic and logical operations such as addition, subtraction, multiplication, logical or and greater-than comparison. The router sends the data coming from neighbouring PEs or the previous ALU results to the ALU for processing. The new ALU results can either be stored temporarily in a register or sent to other PEs depending on the configuration decoded from the instructions. Besides, the router has a buffer-less bypath path that enable data transfer from a PE to a distant PE within one cycle [14], [15]. When a PE transfers data to a distant PE, the PEs along the transfer route are configured to enable the bypass path to allow data to directly pass through it.

In this work, the CGRA is enhanced by introducing a configurable MAC unit with a truncation block, into the PE. The configurable MAC unit works into three modes: MUL, ADD and MAC mode, depending on the opcode decoded from the instructions. The configurable MAC helps to reduce the number of execution cycles when the CGRA processes workloads that have abundant of MAC operations. As discussed earlier, the conventional CGRA takes two instruction cycles to complete a MAC operation as the CGRA PE computes a single operation in a cycle, either MUL or ADD. The configurable MAC is elaborated in Section II-C.

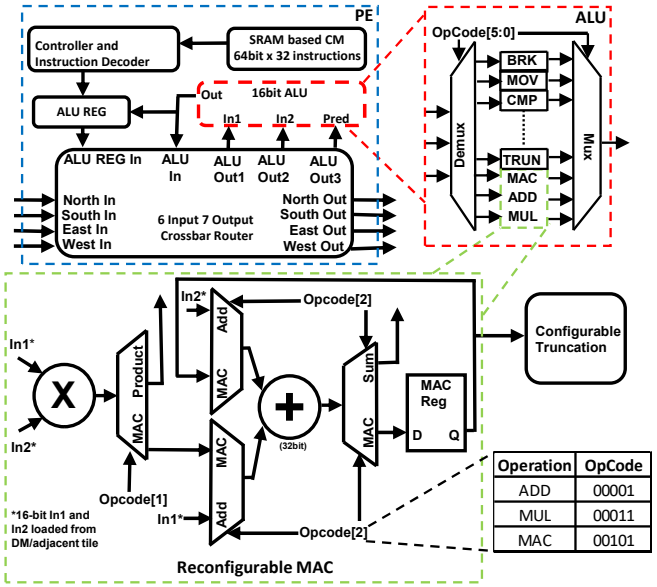

Fig. 3. Proposed CGRA PE with the configurable MAC

C. Configurable MAC

The configurable MAC, as shown in Fig. 3, is a sub-component in the CGRA PE to support three operations: MUL, ADD and MAC, depending on the opcode decoded from the PE instruction. The configurable MAC is designed to reuse the same signed integer multiplier and adder to support all three operations to save hardware resources. Multiplexers (MUXes) are used in the configurable MAC to configure the data path for different operation modes. When performing MUL and ADD, the computation results are sent out from the ALU. When performing MAC operation, the accumulation result is stored in an accumulation register. The bit width of accumulation result is two times larger than the 16-bit input to prevent precision-loss during accumulation. During readout, the accumulation result is passed to the truncation block which is elaborated in Section II-D, to reduce the accumulation result bit width to match with the data bit width of the CGRA.

D. Configurable truncation block

In the proposed CGRA, the MAC accumulation register in the PE is 32-bit wide as the operand data is 16-bit, to avoid precision loss during accumulation. Precision truncation reduces the MAC result to 16-bit, so that it is ready to be used by next PE for further processing or store back to the DM.

In this work, the truncation is performed after the all accumulation cycles are done. As suggested in [16], the accuracy when running neural network inference is higher when applying truncation after MAC as compared to truncating after multiplication. Also, in [16], when handling the overflow situation, the boundary check (BC) mechanism has less impact on the accuracy as compared to the no boundary check (NBC). Hence, the BC mechanism is employed, as illustrated in Fig. 4(a). The BC checks if the overflow bits have one or more ones. If there is, all output bits are set to one, i.e., clipping to maximum value.

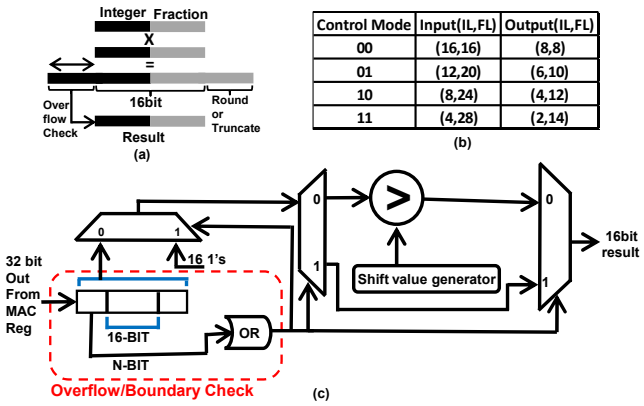

Fig. 4. (a) Boundary check mechanism for overflow detection, (b) 4 configuration modes supported by the (c) configurable truncation block.

As depicted in in Fig. 4(b), the truncation block supports 4 precision modes configured using 2 bits from the instruction. For instance, in mode 00, to output a fixed point number that is of 8-bit integer and 8-bit fractional (FXP$\langle 8, 8\rangle$), the accumulation result is right-shifted by 8 bits and then the lower 16 bits are taken as the outputs. This configurability allows precise control over the truncation and ensures that the appropriate bits are retained according to the desired precision. The other configurations are FXP$\langle 4, 12\rangle$ and FXP$\langle 2, 14\rangle$, since the analysis of CNN accuracy when trained using MNIST dataset and the length of the fractional part for a 16-bit weight in [17] shows that using configurations such as FXP$\langle 4, 12\rangle$ and FXP$\langle 2, 14\rangle$ yields satisfactory accuracy results.

Fig. 4(c) shows the design of the configurable truncation block. It first checks if there is any overflow. If yes, the result is set to all ones to represent the maximum clip value. Else, the integer and fractional part are individually truncated. A separate opcode is allocated to truncate MAC results.

III. CGRA Performance Evaluation

A. Data flow in conventional and proposed PE

During runtime, the PE decodes and executes an instruction read from the CM every clock cycle. The PE loops through the CM until it meets the stopping criterion. The PE sets the ALU and router for the desired operation, and to send/receive data as per the decoded configuration.

In a conventional PE, a MAC operation is performed in two cycles. The operation sequence can be represented as a data flow graph as shown in Fig. 5. The input data is loaded from the DM to the ALU input registers. Then, at time step t_1, a MUL instruction is executed to perform MUL. The product either stays in the same PE or is transfered to the next PE for accumulation. At time step t_2, an ADD instruction is executed to sum the operands taken from the input registers. The addition result is assumed to be stored in a ALU register at the end of t_2. Thus, the CGRA requires two time units and two instruction reads to complete a MAC operation.

In the proposed PE, a MAC operation is performed in one cycle, as shown in Fig. 5. Activities at time t_0 are identical to the conventional PE. Then, the MAC operation occurs at

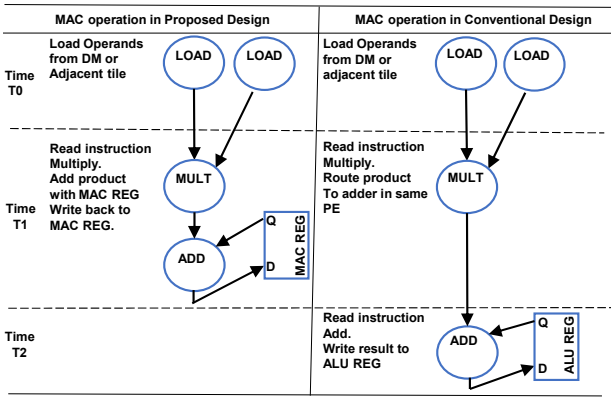

Fig. 5. Data flow of the proposed and conventional PE

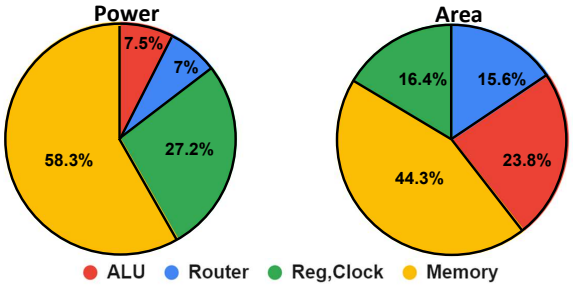

Fig. 6. Power and area breakdown of the proposed PE

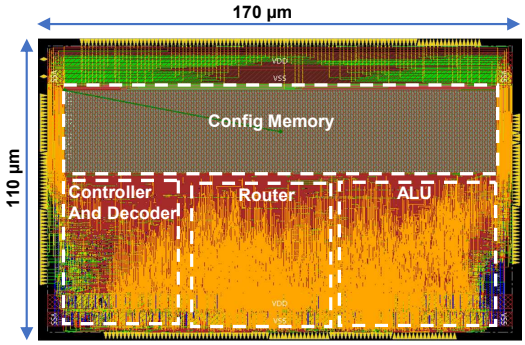

Fig. 7. Layout of the proposed PE

t_1 upon the MAC instruction is executed, where MUL is performed on operands, and the product is accumulated with the previous MAC register results, using the adder within the same PE. The new MAC value is then stored in the accumulation register. Hence, by executing just one instruction in a single cycle, the entire MAC operation can be completed within the PE.

B. Performance of the proposed PE

The proposed PE is implemented using Verilog and synthesized using a 40nm technology node. The total area is 0.019 mm^2. Fig. 6 gives the power and area breakdown of the PE. The layout of the PE is shown in Fig. 7. Although the proposed PE introduces the reconfigurable MAC to complete MAC operation in one cycle, the hardware path delay of the MAC operation is roughly 3.2ns, which is 1.86 times longer the path delay of the multiplier. Despite the increase in path delay, the proposed CGRA can still meet the target operating frequency of 100MHz in this work.

TABLE I
COMPARISON WITH STATE-OF-THE-ART CGRA

	Amber [19]	SSCL'20 [20]	ISSCC'19 [21]	TVLSI'18 [22]	ASSCC'19 [14]	JSSC'20 [23]	**Proposed**
Tech (nm)	16	28	22	55	40	28	**40**
Voltage (V)	0.84	0.6	0.8	NA	1.1	0.9	**1**
Frequency (MHz)	955	89	36	450	853	800	**100**
PE count	384	120	15	30	16	64	**64**
Throughput (GOPS)	367	14.1	145	77.4	6.5	0.9	**9.5**
Power (mW)	NA	45.9	NA	1526	6.5	537	**47.6**
Efficiency (GOPS/W)	538@0.84V	307@0.6V	978@0.48V	50.8	90	196	**199.2@1V 420.7@0.6V**
Norm. Efficiency (GOPS/W) [1]	86	150.4	295.8	96	90	96	**420.7**

[1] Norm. Efficiency = Efficiency $\times(\frac{\text{node}}{40\text{nm}})^2$, derived using general scaling in Table 1 from [24] for coarse performance estimation

The power and energy consumption of the proposed CGRA PE is analyzed when processing only the MAC operations, as compared to the conventional PE. Both conventional and proposed PE are simulated with the GeMM workload and their power consumption are obtained using Cadence Joules. For instance, in an 8x8 GeMM workload, the average power consumption of the proposed CGRA PE is 0.74mW when processing 512 MAC operations, which is similar to the power consumption of the conventional PE. However, the energy consumption of the proposed PE is 0.34 times smaller than the conventional PE, when processing the same number of MAC operations for 8x8 GeMM. As shown in Fig. 5, the proposed PE only requires one memory access for one MAC operation, yet, the conventional PE requires 2 memory accesses to obtain the instructions to complete one MAC operation. Based on our simulation, the energy consumption of CM read access is 5.14 and 2.19 times than that of ADD and MUL respectively. Thus, the energy savings of the proposed PE is 37%.

C. Performance of the CGRA

To evaluate the performance impact of the proposed PE on the CGRA performance, the CGRA is simulated using the GeMM workload. GeMM serves as a foundational computation in numerous applications such as speech processing and fast Fourier transform [18]. It is a well-known fundamental operation in linear algebra and is commonly employed as a benchmark to assess the performance of high-performance computing systems [18]. It holds a significant position in the 13-dwarfs categorization of computational patterns [10]. The equation of GeMM is given in Equation 1. The proposed CGRA utilizes the compiler proposed in [7] to compile the GeMM workload.

$$A[\,] = B[\,] \times C[\,] \tag{1}$$

where $A[\,]$ is the resultant matrix, $B[\,]$ and $C[\,]$ are the input matrices.

Fig. 8 shows the energy saved by the proposed CGRA when running GeMM workload. The energy savings are mainly contributed by the reduction in instructions read from the CM in the PEs. Considering the 8x8 GeMM workload as an example, the conventional CGRA requires 512 MUL and 448 ADD, totaling to 960 arithmetic instruction reads from the CM. For the proposed CGRA, this number reduces to only 512 MACs, which is a 54% drop. However, the total instructions generated

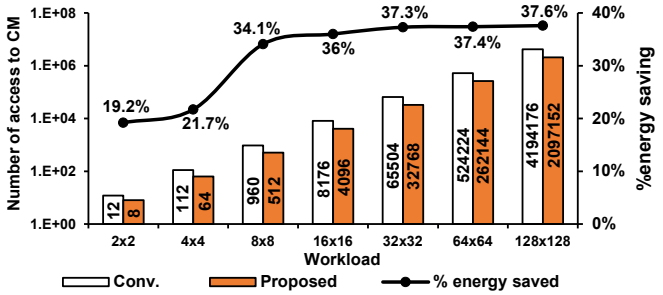

Fig. 8. Number of instruction reads and energy savings when the proposed CGRA runs various GeMM workloads as compared to conventional CGRA

by the compiler are more than 512, due to the overhead of control instructions for address generation for input and output data access from the DM. This effectively results in one MAC performed in 1.35 clock cycles. Overall, the throughput achieved by the proposed CGRA is 9.5GOPS, consuming an average power of 47.6mW at 1V supply. The peak efficiency observed is 420.7GOPS/W at 0.6V and 21MHz.

Table I compares the proposed CGRA with the state-of-the-art. The efficiency is normalized to 40nm technology node for fair comparison. The proposed CGRA attains peak efficiency of 420.7 GOPS/W, operating at 0.6V and 21MHz, which is about 1.4 times higher than [21]. The design from [21] comes closest to this work, achieving high energy efficiency due to distributed memories at the lowest hierarchy level, high PE count and multi-directional routing.

IV. CONCLUSION

A CGRA that employs PEs using the proposed configurable MAC with truncation block has been presented. The configurable MAC block allows the CGRA to complete a MAC operation in one cycle, reducing the number of processing cycles, whilst reusing the same hardware for MUL and ADD. The configurable truncation block is included to allow users to choose the precision when the MAC result accumulated at full precision leaves the PE. The proposed CGRA, synthesized using a 40nm technology node, attains efficiency of 420.6GOPS/W operating at supply of 0.6V and frequency of 21MHz, which is 1.4 times higher than the state-of-the-art.

ACKNOWLEDGEMENT

This research is supported by the National Research Foundation, Singapore (CRP23-2019-0003).

REFERENCES

[1] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.

[2] G. Moore, "Cramming more components onto integrated circuits," *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, 1998.

[3] T. N. Theis and H.-S. P. Wong, "The end of moore's law: A new beginning for information technology," *Computing in Science & Engineering*, vol. 19, no. 2, pp. 41–50, 2017.

[4] B. Degnan, B. Marr, and J. Hasler, "Assessing trends in performance per watt for signal processing applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 1, pp. 58–66, 2016.

[5] Z. Li, D. Wijerathne, and T. Mitra, "Coarse-Grained Reconfigurable Array (CGRA)," in *Handbook of Computer Architecture*. Springer, Singapore, 2023, pp. 1–41.

[6] D. Wijerathne, Z. Li, T. K. Bandara, and T. Mitra, "PANORAMA: divide-and-conquer approach for mapping complex loop kernels on CGRA," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*. ACM, Jul. 2022, pp. 127–132.

[7] D. Wijerathne, Z. Li, M. Karunaratne, L.-S. Peh, and T. Mitra, "Morpher: An Open-Source Integrated Compilation and Simulation Framework for CGRA," *Fifth Workshop on Open-Source EDA Technology (WOSET)*, Nov. 2022.

[8] Aliagha, Ensieh and Gohringer, Diana, "Energy Efficient Design of Coarse-Grained Reconfigurable Architectures: Insights, Trends and Challenges," in *2022 International Conference on Field-Programmable Technology (ICFPT)*, Dec. 2022, pp. 1–11.

[9] L. Liu, J. Zhu, Z. Li, Y. Lu, Y. Deng, J. Han, S. Yin, and S. Wei, "A survey of coarse-grained reconfigurable architecture and design: Taxonomy, challenges, and applications," *ACM Comput. Surv.*, vol. 52, no. 6, oct 2019. [Online]. Available: https://doi.org/10.1145/3357375

[10] J. Lee and J. Lee, "Np-cgra: Extending cgras for efficient processing of light-weight deep neural networks," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021, pp. 1408–1413.

[11] E. Adams, S. Venkatachalam, and S.-B. Ko, "Energy-efficient approximate mac unit," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–4.

[12] P. Jagadeesh, S. Ravi, and K. H. Mallikarjun, "Design of high performance 64 bit mac unit," in *2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT)*, 2013, pp. 782–786.

[13] K. J. M. Martin, "Twenty years of automated methods for mapping applications on cgra," in *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2022, pp. 679–686.

[14] B. Wang, M. Karunarathne, A. Kulkarni, T. Mitra, and L.-S. Peh, "Hycube: A 0.9v 26.4 mops/mw, 290 pj/op, power efficient accelerator for iot applications," in *2019 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, 2019, pp. 133–136.

[15] M. Karunaratne, A. K. Mohite, T. Mitra, and L. S. Peh, "HyCUBE: A CGRA with Reconfigurable Single-cycle Multi-hop Interconnect," *Proceedings - Design Automation Conference*, vol. Part 128280, Jun. 2017.

[16] F. Taheri, S. B. Sarmadi, H. Mosanaei-Boorani, and R. Taheri, "Fixflow: A framework to evaluate fixed-point arithmetic in light-weight CNN inference," *CoRR*, vol. abs/2302.09564, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2302.09564

[17] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, p. 1737–1746.

[18] A. Pedram, J. McCalpin, and A. Gerstlauer, "Transforming a linear algebra core to an fft accelerator," in *2013 IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors*, 2013, pp. 175–184.

[19] A. Carsello *et al.*, "Amber: A 367 gops, 538 gops/w 16nm soc with a coarse-grained reconfigurable array for flexible acceleration of dense linear algebra," *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, pp. 70–71, 2022.

[20] S. Smets, M. D. Gomony, M. Jivanescu, T. Goedemé, and M. Verhelst, "A 28-nm coarse grain 2d-reconfigurable array with data forwarding," *IEEE Solid-State Circuits Letters*, vol. 3, pp. 226–229, 2020.

[21] S. Smets, T. Goedemé, A. Mittal, and M. Verhelst, "2.2 a 978gops/w flexible streaming processor for real-time image processing applications in 22nm fdsoi," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, 2019, pp. 44–46.

[22] X. Fan, D. Wu, W. Cao, W. Luk, and L. Wang, "Stream processing dual-track cgra for object inference," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 6, pp. 1098–1111, 2018.

[23] G. Peng, L. Liu, S. Zhou, S. Yin, and S. Wei, "A 2.92-gb/s/w and 0.43-gb/s/mg flexible and scalable cgra-based baseband processor for massive mimo detection," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 2, pp. 505–519, 2020.

[24] A. Stillmaker and B. Baas, "Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm," *Integration*, vol. 58, pp. 74–81, Jun. 2017, publisher: Elsevier B.V.